

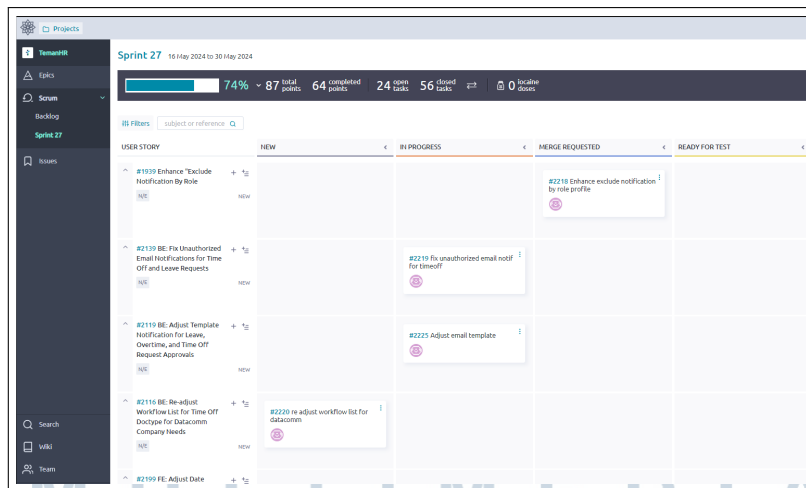
BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama kerja magang di PT Sumber Inovasi Informatika, posisi yang ditempati adalah sebagai *backend software developer* dengan CTO sebagai *supervisor* selama kerja magang. *backend software developer* bekerja sebagai penghubung antara pengguna aplikasi dengan data, biasanya melalui penggunaan *endpoint*. Posisi ini perlu berinteraksi dengan *frontend software developer* untuk mencegah terjadinya kesalahan komunikasi.

Adapun proses koordinasi dalam suatu tim *software developer*. Pertama, *Product Owner* akan membentuk tiket di sebuah situs bernama *Taiga*, tiket tersebut memiliki *agile user story* untuk mempermudah pemahaman tujuan suatu tiket. Kemudian tiket-tiket tersebut dibagikan kepada para *software developer* untuk dikerjakan. Setelah menyelesaikan tiket yang sudah diberi, hasil pekerjaan diunggah ke situs *GitLab*, yang kemudian diuji oleh *product owner* untuk memastikan bahwa fitur sudah sesuai dengan *user requirement*. Gambar 3.1 menunjukkan halaman situs *Taiga*.

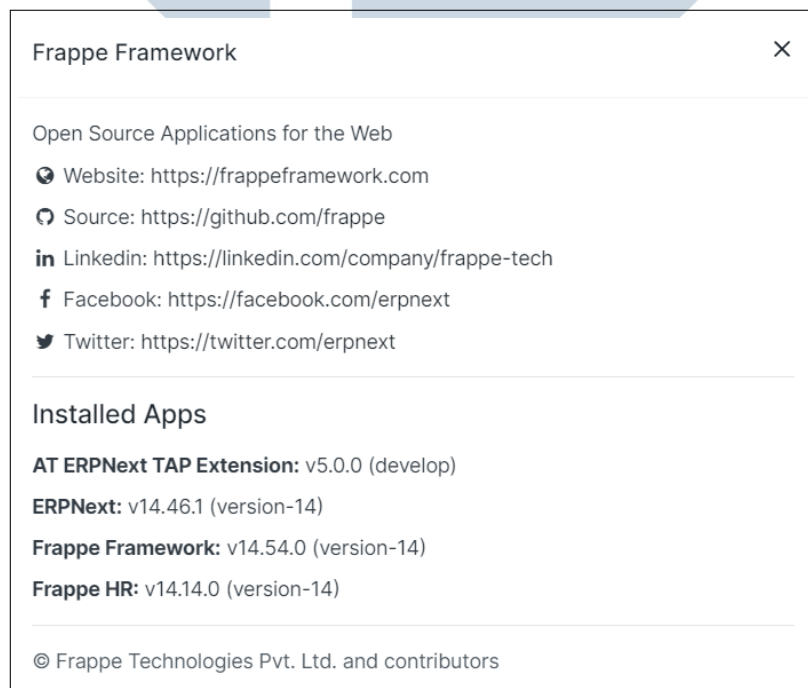


Gambar 3.1. Halaman Situs Taiga

Dalam konteks kolaborasi antara *backend software developer* dan *frontend software developer*, terdapat proses yang terstruktur. Apabila terdapat kebutuhan fitur aplikasi yang memerlukan *API* khusus, *frontend software developer* akan meminta *backend software developer* untuk merancang dan mengimplementasikan

API tersebut. *Frontend software developer* akan menyediakan spesifikasi yang diperlukan, termasuk informasi dan data yang relevan, untuk memastikan fitur tersebut dapat diimplementasikan. Selanjutnya, *backend software developer* akan merancang *API* yang menghasilkan respons *JSON* yang sesuai dengan spesifikasi yang diberikan oleh *frontend software developer*. Ini mencerminkan kerjasama yang efisien dan profesional dalam tim *software developer*.

Selama kerja magang di PT Sumber Inovasi Informatika, *Framework* yang digunakan adalah *framework Frappe*, sebuah *framework* aplikasi yang kuat dan ditulis dalam *Python* dan *Javascript*. Sebagai *framework open-source*, *Frappe* memberikan kebebasan bagi *developer* untuk menyesuaikan dan memperluas fungsionalitasnya sesuai dengan kebutuhan proyek. *Framework Frappe* memberikan sejumlah fitur canggih yang memungkinkan *developer* untuk mengurangi penulisan kode secara signifikan [5]. Gambar 3.2 menunjukkan bagian dari halaman web yang berkaitan dengan *Framework Frappe* serta aplikasi web yang digunakan pada proyek.



Gambar 3.2. Tentang Framework Frappe

Adapun beberapa pertemuan yang wajib dilaksanakan sebagai *backend software developer* pada tim yaitu sebagai berikut.

1. Stand Up Meeting

Stand up meeting adalah pertemuan dimana setiap anggota pada tim berpartisipasi untuk membagikan apa yang telah dikerjakan pada hari kemarin dan apa yang akan dikerjakan pada hari itu. Pertemuan ini memerlukan semua anggota tim untuk berdiri pada saat dilaksanakan untuk memberikan ketidaknyamanan ketika Pertemuan sudah terlalu lama. Tujuannya agar pertemuan memiliki waktu yang singkat.

2. Sprint Planning

Sprint planning meeting adalah pertemuan dimana *product owner* menjelaskan tiket yang akan dikerjakan selama satu *sprint* ke depan. Tiket tersebut akan di nilai oleh setiap anggota tim berdasarkan tingkat kesulitannya, seberapa lama dikerjakan, dan seberapa susah dikerjakan. Tujuan dari pertemuan ini adalah untuk memberikan gambaran dan mempersiapkan tim untuk melanjutkan ke *sprint* selanjutnya.

3. Retro Meeting

Retro meeting adalah pertemuan dimana setiap anggota pada tim berpartisipasi untuk membagikan apa yang membuat anggota tersebut senang, sedih, dan bingung pada saat melaksanakan *sprint*. Tujuan dari pertemuan ini adalah untuk melakukan penyesuaian dalam tim agar setiap anggota merasa nyaman selama melaksanakan *sprint*.

3.2 End-to-End Software Development Team

Selama kerja magang di PT Sumber Inovasi Informatika, setiap proyek dikelola oleh tim yang ditugaskan khusus. Anggota tim ini memiliki peran spesifik yang berbeda-beda, namun meski demikian, setiap peran dalam tim saling bergantung satu sama lain. Ini menunjukkan bahwa kolaborasi dan kerja sama tim adalah hal yang sangat penting dalam pengembangan proyek.

3.2.1 Tim Proyek TemanHR

Pada tim *TemanHR*, terdapat empat peran penting yaitu sebagai berikut.

1. Product Owner

Dalam tim *TemanHR*, *product owner* adalah peran yang menerima permintaan *Stakeholder*. Permintaan tersebut kemudian dijadikan tiket dalam situs *Taiga* yang kemudian diberikan kepada tim *developer*. Ketika tiket sudah selesai dikerjakan, maka *product owner* akan melakukan *testing* pada tiket tersebut. *Testing* dilakukan dengan cara menggunakan fitur yang telah dibuat dan menyesuaikan dengan *user requirementnya*. Jika tiket sudah sesuai dan tidak ada revisi, maka tiket akan dimasukkan kedalam kelompok "*test passed*".

2. UI/UX

Dalam tim *TemanHR*, *UI/UX* adalah peran yang merancang desain tampilan *mobile*. Peran ini bekerja sama dengan *frontend software developer* dengan mengkomunikasikan perancangan tampilan *mobile* agar saling memahami perancangan tampilan bersama. Peran ini menggunakan *Figma* untuk perancangan tampilan *mobile* dan juga pembuatan *icon* yang diperlukan dalam aplikasi *TemanHR*.

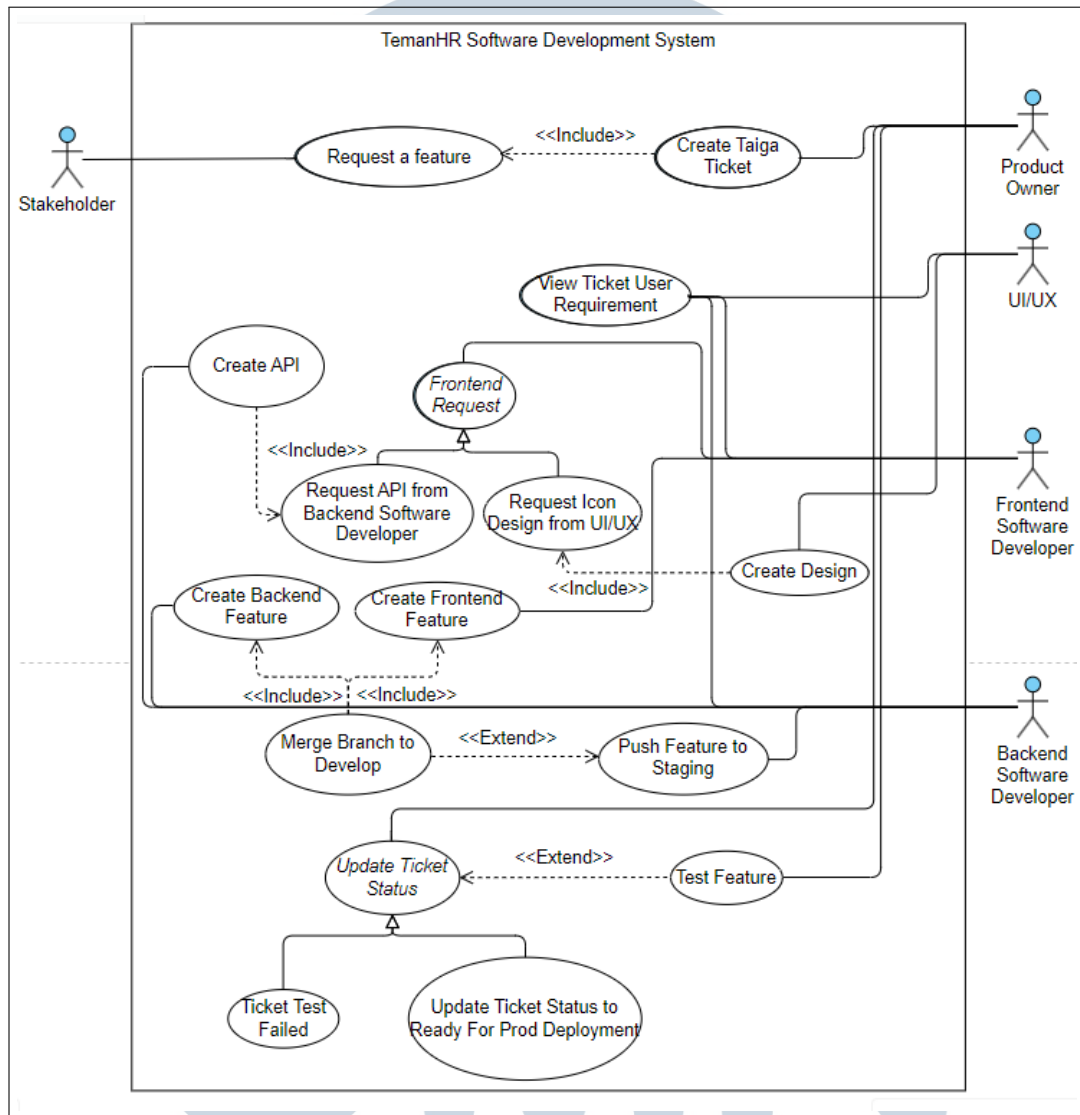
3. Frontend Software Developer

Dalam tim *TemanHR*, *frontend software developer* adalah peran yang mengimplementasikan antarmuka pengguna. Peran ini memerlukan bantuan dari *UI/UX* untuk perancangan desain dan *backend software developer* untuk perancangan *API*. Peran ini menggunakan aplikasi *Android Studio* dan *framework React js* untuk perancangan aplikasi *TemanHR*.

4. Backend Software Developer

Dalam tim *TemanHR*, *backend software developer* adalah peran yang merancang *API* dan juga fungsionalitas yang diperlukan untuk mendukung aplikasi *TemanHR*. Peran ini bekerja sama dengan *frontend software developer* dengan mengkomunikasikan *respons* dan *request* untuk membentuk *API* yang cocok untuk digunakan *frontend software developer*. Peran ini menggunakan aplikasi *Visual Studio Code* atau *PyCharm* dan *framework Frappe*.

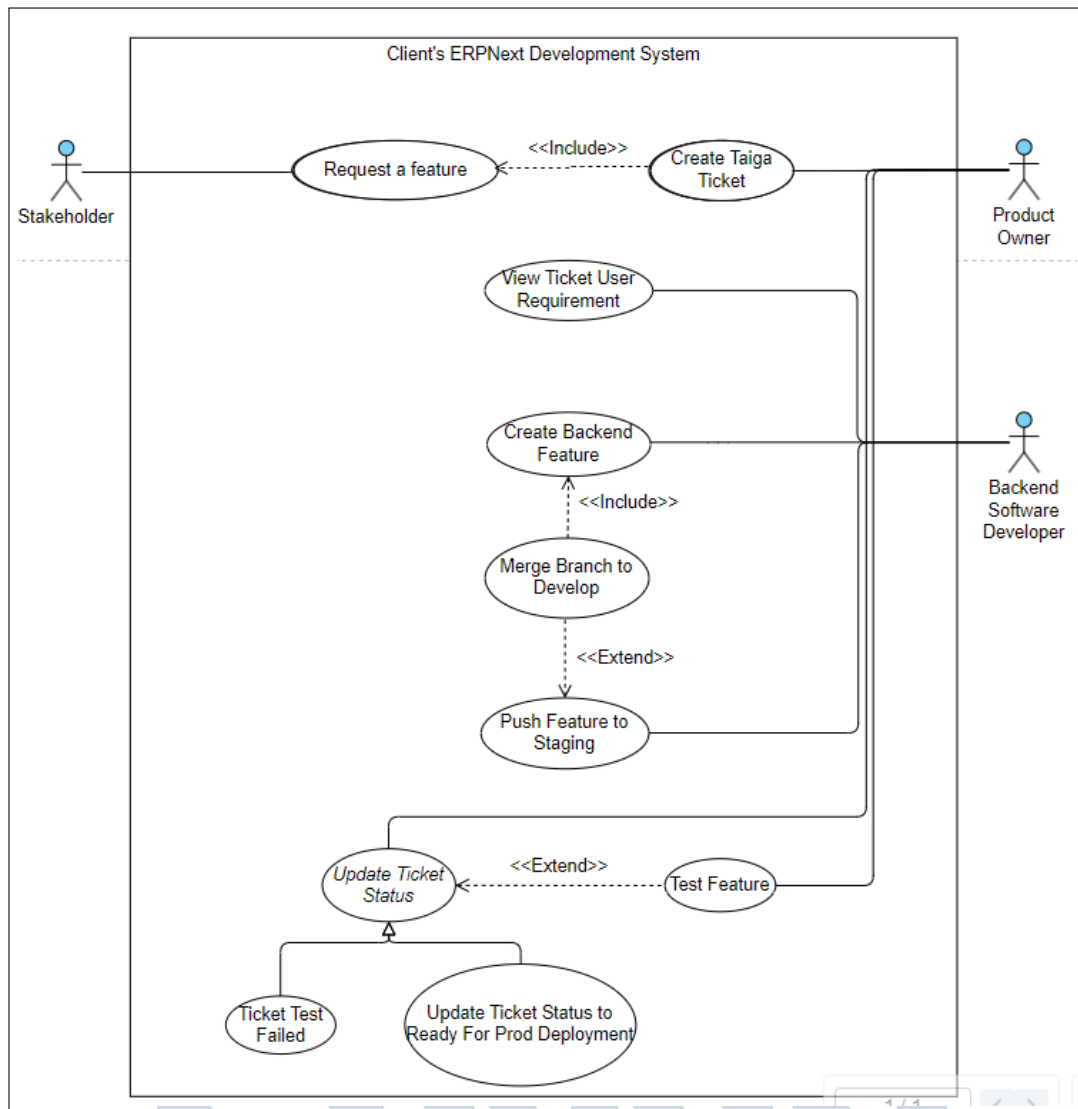
Gambar 3.3 menunjukkan diagram *use case* yang mengilustrasikan sistem *software development* dalam tim *TemanHR*.



Gambar 3.3. Use Case TemanHR Software Development System

3.2.2 Tim Proyek Sistem ERPNext Klien

Pada tim sistem *ERPNext* klien, terdapat dua peran penting yaitu sebagai berikut.



Gambar 3.4. Use Case Client's ERPNext Development System

1. Product Owner

Dalam tim sistem *ERPNext* klien, *product owner* adalah peran yang menerima permintaan klien. Permintaan tersebut kemudian dijadikan tiket dalam situs *Taiga* yang kemudian diberikan kepada tim *developer*. Ketika tiket sudah selesai dikerjakan, maka *product owner* akan melakukan *testing* pada tiket tersebut. *Testing* dilakukan dengan cara menggunakan fitur yang telah dibuat dan menyesuaikan dengan *user requirement*nya. Jika tiket sudah sesuai dan tidak ada revisi, maka tiket akan dimasukkan kedalam kelompok "*test passed*".

2. Backend Software Developer

Dalam tim sistem *ERPNext* klien, *backend software developer* adalah peran yang merancang fungsionalitas setiap tiket yang dibentuk oleh *product owner*. Peran ini memastikan bahwa program berjalan sesuai dengan *user requirement* yang sudah ditentukan oleh *product owner*.

Gambar 3.4 menunjukkan diagram *use case* yang mengilustrasikan sistem *ERPNext development* dalam tim *ERPNext* klien.

3.3 Tugas yang Dilakukan

Selama kerja magang di PT Sumber Inovasi Informatika, ada tugas yang harus dilakukan sebagai *backend software developer*. Aktivitas yang dilakukan selama kerja magang dapat dilihat pada tabel 3.1.



Tabel 3.1. Tugas yang Dilakukan Selama Masa Magang

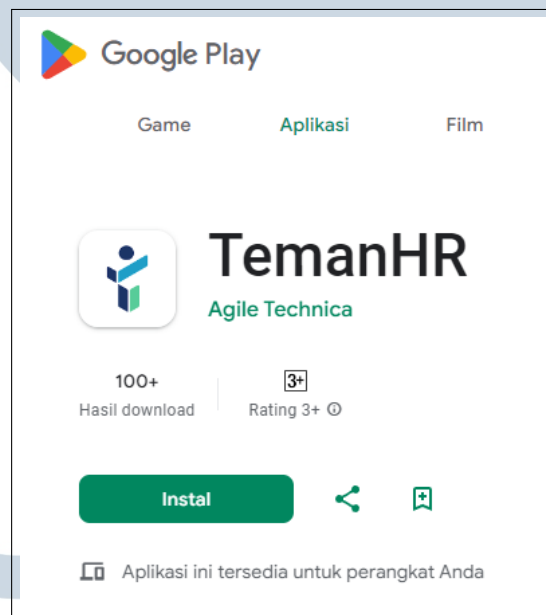
Minggu Ke-	Pekerjaan yang dilakukan
1	Persiapan <i>software</i> dan belajar <i>framework Frappe</i> .
2	Masuk ke dalam proyek pengembangan aplikasi <i>TemamHR</i> .
3	Pembuatan fitur validasi minimal jam kerja untuk aplikasi <i>TemamHR</i> .
4	Melakukan investigasi terhadap kemunculan proyek-proyek lama yang sudah tidak aktif saat melakukan pencarian melalui beberapa fitur.
5	Pembuatan format penamaan seri yang berurutan untuk lampiran pada dokumen <i>expense claim</i> .
6	Pembuatan <i>workflow</i> untuk dokumen <i>overtime</i> .
7	Pembuatan fitur pembatasan user yang bisa melihat dan membuka fitur pengajuan pada aplikasi <i>TemamHR</i> .
8	Masuk ke dalam proyek pengembangan sistem <i>ERPNext</i> untuk klien.
9	Pembuatan fitur notifikasi pengingat absensi pada aplikasi <i>TemamHR</i> .
10	Melakukan investigasi dokumen <i>sales order</i> dan persiapan <i>UAT (User Acceptance Testing)</i> untuk penutupan fase proyek.
11	Melakukan pengecekan dokumen-dokumen yang akan di gunakan untuk <i>UAT</i> .
12	Melaksanakan <i>UAT</i> di kantor klien.
13	Pembuatan fitur notifikasi pengingat absensi untuk karyawan yang memiliki jam kerja khusus pada aplikasi <i>TemamHR</i> .
14	Melakukan investiasi untuk kehilangan beberapa dokumen <i>timesheet</i> pada <i>TemamHR</i> .
15	Melakukan investigasi perbedaan total jam kerja pada aplikasi <i>TemamHR</i> dan situs <i>Frappe</i> .
16	Mengerjakan dan menyesuaikan beberapa fitur untuk menyelesaikan fase satu klien.

3.4 Uraian Pelaksanaan Magang

Selama kerja magang di PT Sumber Inovasi Informatika, ada berbagai macam fitur yang perlu dibuat sebagai *backend software developer*. Fitur-fitur yang dibuat yaitu sebagai berikut.

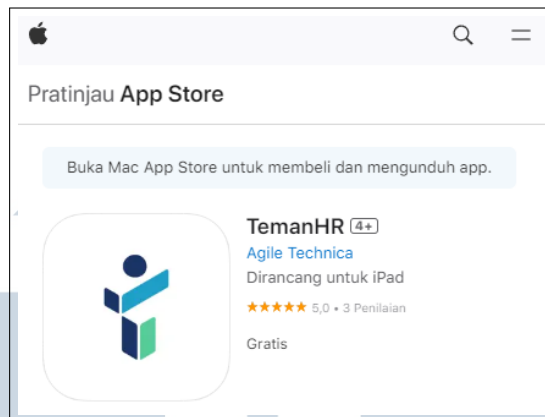
3.4.1 Pengembangan TemanHR

TemanHR adalah aplikasi *mobile* yang dibuat oleh PT Sumber Inovasi Informatika. Aplikasi ini memudahkan proses *HR* dan juga penggajian. Karyawan yang menggunakan aplikasi ini dapat melakukan absensi dan mencatat kegiatan harian dengan mudah dan efisien. Aplikasi ini tersedia untuk *Android* dan *IOS*. Gambar 3.5 dan 3.6 menunjukkan aplikasi *TemanHR* di *Google Playstore* dan *App Store*.



Gambar 3.5. Aplikasi TemanHR di Google Playstore

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.6. Aplikasi TemanHR di App Store

Saat bergabung dengan proyek TemanHR, tahap yang sedang ditempuh adalah tahap development. Kontribusi yang sudah diberikan kepada proyek ini adalah sebagai berikut.

A. Fitur Validasi Minimal Jam Kerja Sebelum Absen Pulang

Fitur validasi minimal jam kerja sebelum absen pulang adalah proses pengecekan untuk memastikan bahwa karyawan sudah bekerja selama durasi waktu tertentu sebelum absen pulang. Fitur ini akan berjalan pada saat karyawan yang menggunakan TemanHR ingin melakukan absen pulang dengan menekan tombol *tap out*. Jika karyawan belum memenuhi jam kerja yang sudah ditentukan oleh perusahaan, maka pada saat tombol *tap out* ditekan, akan keluar pop up tidak bisa absen pulang. Tujuan dari fitur ini adalah untuk memastikan bahwa karyawan mematuhi standar jam kerja yang sudah ditetapkan oleh perusahaan. Gambar 3.7 adalah bukti bahwa fitur ini telah mendapatkan persetujuan dari *product owner* dan sudah sesuai dengan *user requirement* yang sudah ditentukan sebelumnya.



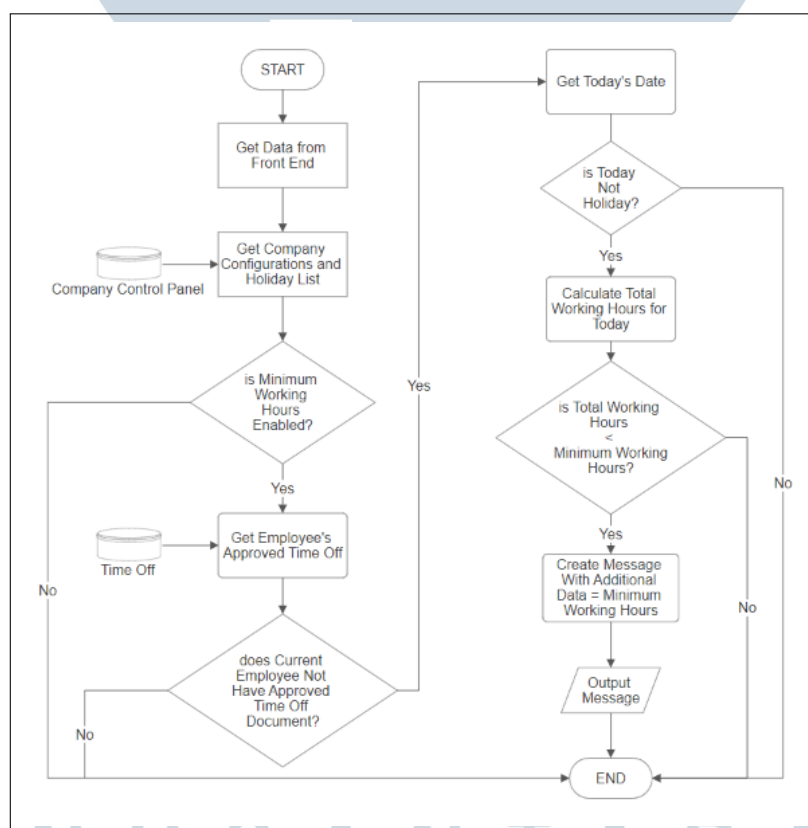
Gambar 3.7. Persetujuan Fitur Validasi Minimal Jam Kerja Sebelum Absen Pulang

1. User Requirement

Adapun *user requirement* untuk fitur validasi minimal jam kerja sebelum absen pulang yaitu sebagai berikut.

- (a) *User* yang berwenang harus dapat mengatur durasi minimal jam kerja.
- (b) Sistem harus mengeluarkan pesan informatif seperti minimal jam kerja ketika belum bisa absen pulang.
- (c) Sistem harus bisa memvalidasi apakah karyawan telah bekerja selama durasi minimal jam kerja yang telah di tentukan.
- (d) Sistem harus memiliki kemampuan adaptif, yang berarti jika karyawan memiliki dokumen *time off* yang sedang berlaku atau berada di hari libur, sistem akan melakukan penyesuaian sesuai kondisi tersebut.

2. Flowchart



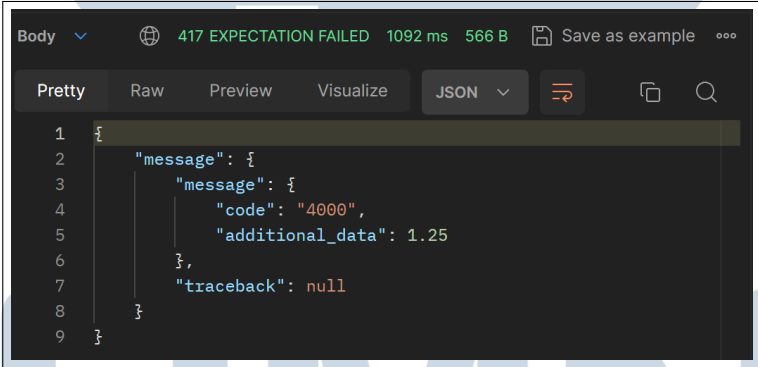
Gambar 3.8. Flowchart Validasi Minimal Jam Kerja

Gambar 3.8 merupakan *flowchart* yang mencakup beberapa proses. Dimulai dengan pengambilan data dari *request* yang dikirimkan oleh *frontend*

software developer, diikuti oleh pengambilan data pengaturan perusahaan karyawan dari *database employee*. Selanjutnya, sistem melakukan pengecekan. Pertama, apakah fitur minimal jam kerja sebelum absen pulang diaktifkan dalam pengaturan perusahaan. Kedua, apakah karyawan memiliki dokumen *time off* yang disetujui, jika ya, maka karyawan tidak akan memiliki minimal jam kerja. Ketiga, apakah hari ini adalah hari libur, jika ya, maka karyawan juga tidak akan memiliki minimal jam kerja.

Setelah itu, sistem melakukan kalkulasi jam kerja karyawan dan membandingkannya dengan minimal jam kerja. Jika jam kerja karyawan melebihi minimal jam kerja, maka karyawan dapat melakukan absen pulang. Namun, jika jam kerja masih kurang dari minimal jam kerja, sistem akan mengirimkan respons ke *frontend software developer*.

3. Implementasi



```
Body 417 EXPECTATION FAILED 1092 ms 566 B Save as example
Pretty Raw Preview Visualize JSON
1 {
2   "message": {
3     "message": {
4       "code": "4000",
5       "additional_data": 1.25
6     },
7     "traceback": null
8   }
9 }
```

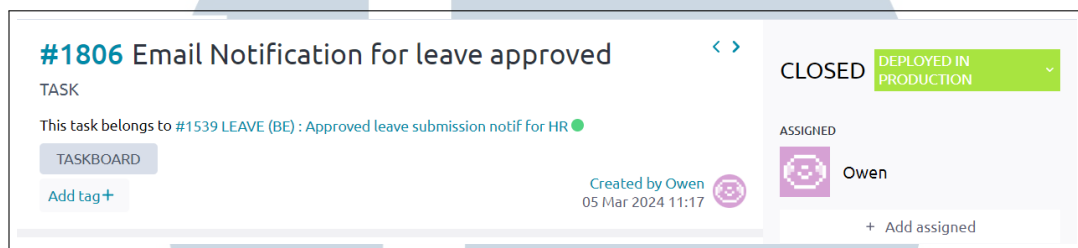
Gambar 3.9. JSON Validasi Minimal Jam Kerja

Gambar 3.9 merupakan *JSON* yang mengirimkan dua pesan ke *frontend software developer*. Pesan “code” merujuk ke file “error_code.txt” untuk deskripsi respons, kode 4000 artinya pesan bisa berbahasa Inggris atau Indonesia tergantung preferensi pengguna. Sementara itu, “additional_data” menyampaikan data dari *database* tentang minimal jam kerja perusahaan.

B. Fitur Notifikasi Terkait Dokumen Cuti yang Perlu Dilihat oleh Departemen HR

Fitur notifikasi terkait dokumen cuti yang perlu dilihat oleh departemen *HR* adalah proses pengiriman notifikasi berupa *email* yang bertujuan untuk memberi

informasi terkait dokumen cuti yang perlu dilihat oleh karyawan *HR*. *Email* notifikasi akan dikirim otomatis pada saat ada dokumen cuti baru. Tujuan dari fitur ini adalah untuk meningkatkan kesadaran karyawan *HR* atas keberadaan dokumen yang baru. Gambar 3.10 adalah bukti bahwa fitur ini telah mendapatkan persetujuan dari *product owner* dan sudah sesuai dengan *user requirement* yang sudah ditentukan sebelumnya.



Gambar 3.10. Persetujuan Fitur Notifikasi Terkait Dokumen Cuti yang Perlu Dilihat oleh Departemen HR

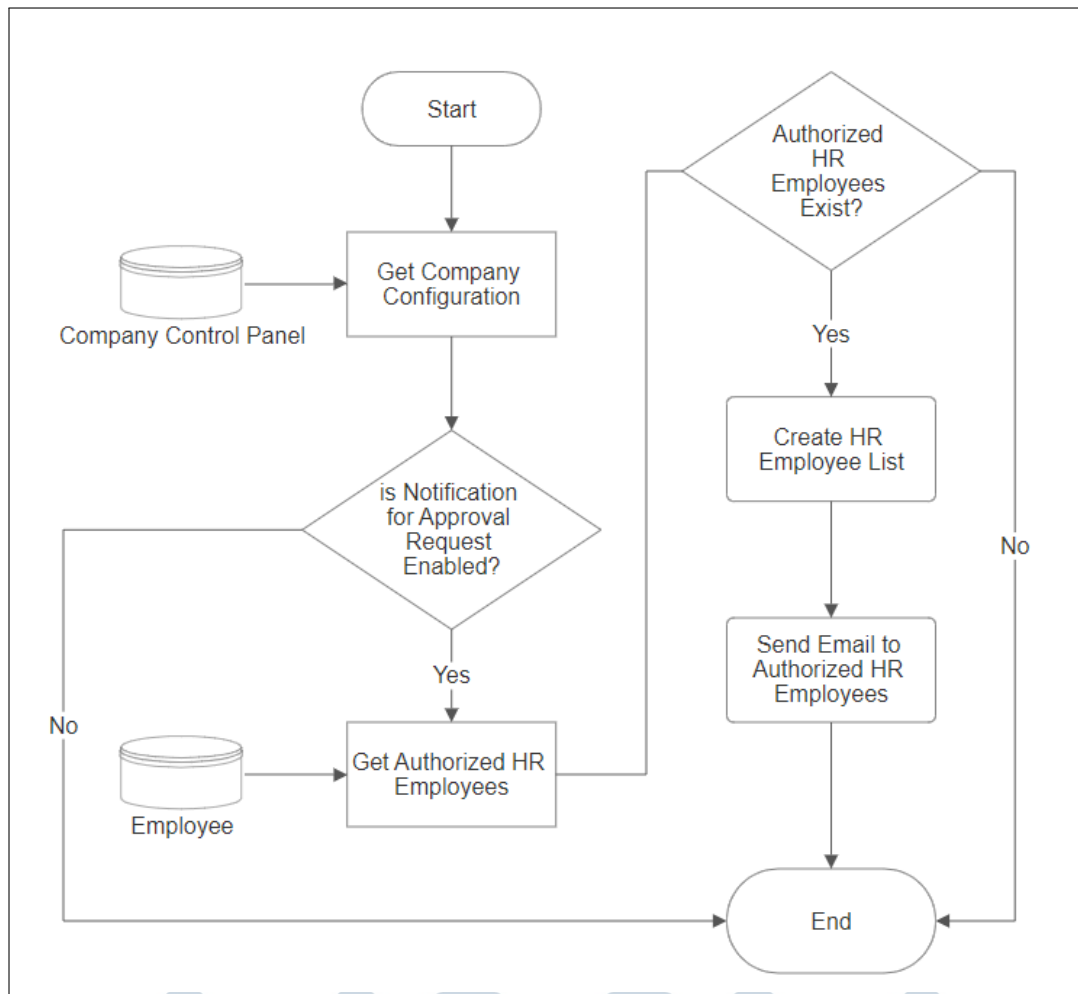
1. User Requirement

Adapun *user requirement* untuk fitur notifikasi terkait dokumen cuti yang perlu dilihat oleh departemen *HR* yaitu sebagai berikut.

- (a) Sistem harus mengirim notifikasi otomatis setiap kali ada pengajuan dokumen cuti yang baru.
- (b) *email* harus memiliki format yang jelas dan mencakup detail penting seperti nama karyawan, tanggal cuti, dan alasan cuti.
- (c) *Email* harus memiliki tombol yang mencakup link langsung ke dokumen cuti yang ditinjau.

2. Flowchart

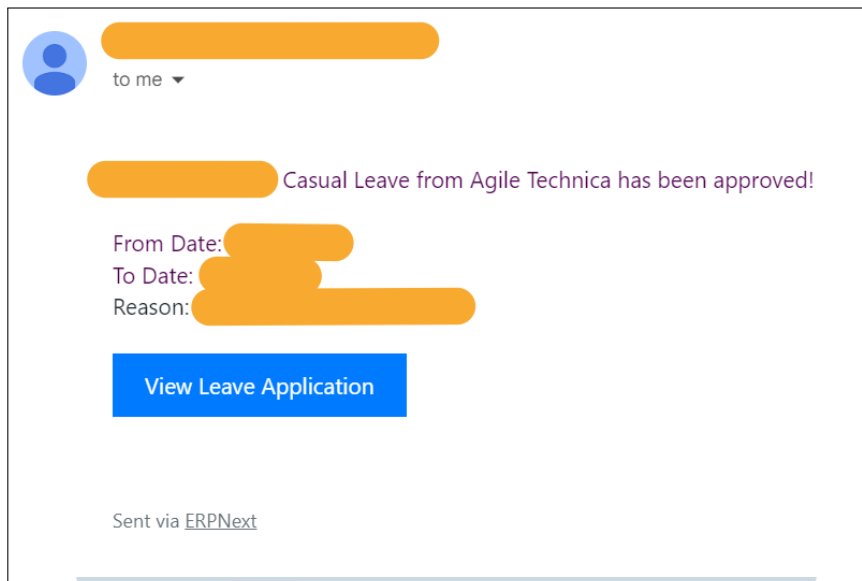
Gambar 3.11 merupakan *flowchart* yang mencakup beberapa proses. Dimulai dengan pengambilan pengaturan perusahaan karyawan dan daftar karyawan *HR* yang berwenang. Selanjutnya dilakukan pengecekan apakah daftar tersebut kosong. Jika iya, maka akan membuat dan mengirimkan *email* kepada para karyawan *HR* yang berwenang.



Gambar 3.11. Flowchart Notifikasi Email Dokumen Cuti

3. Implementasi

Gambar 3.12 menunjukkan tampilan *email* yang diterima oleh karyawan *HR* dari fitur notifikasi terkait dokumen cuti yang perlu dilihat oleh departemen *HR*. Informasi yang disediakan mencakup durasi dan tanggal cuti, alasan untuk cuti, serta sebuah tombol yang mengarahkan ke dokumen cuti tersebut di situs *Frappe*.



Gambar 3.12. Notifikasi Email Dokumen Cuti

C. Fitur Akses Pengajuan Mobile

Fitur akses pengajuan *mobile* adalah pembagian akses kepada karyawan *HR* tertentu yang memberikan kemampuan untuk melakukan persetujuan atau penolakan suatu dokumen permintaan dari karyawan. Tujuan dari fitur ini adalah untuk memberikan akses hanya untuk karyawan tertentu. Gambar 3.13 adalah bukti bahwa fitur ini telah mendapatkan persetujuan dari *product owner* dan sudah sesuai dengan *user requirement* yang sudah ditentukan sebelumnya.



Gambar 3.13. Persetujuan Fitur Akses Pengajuan Mobile

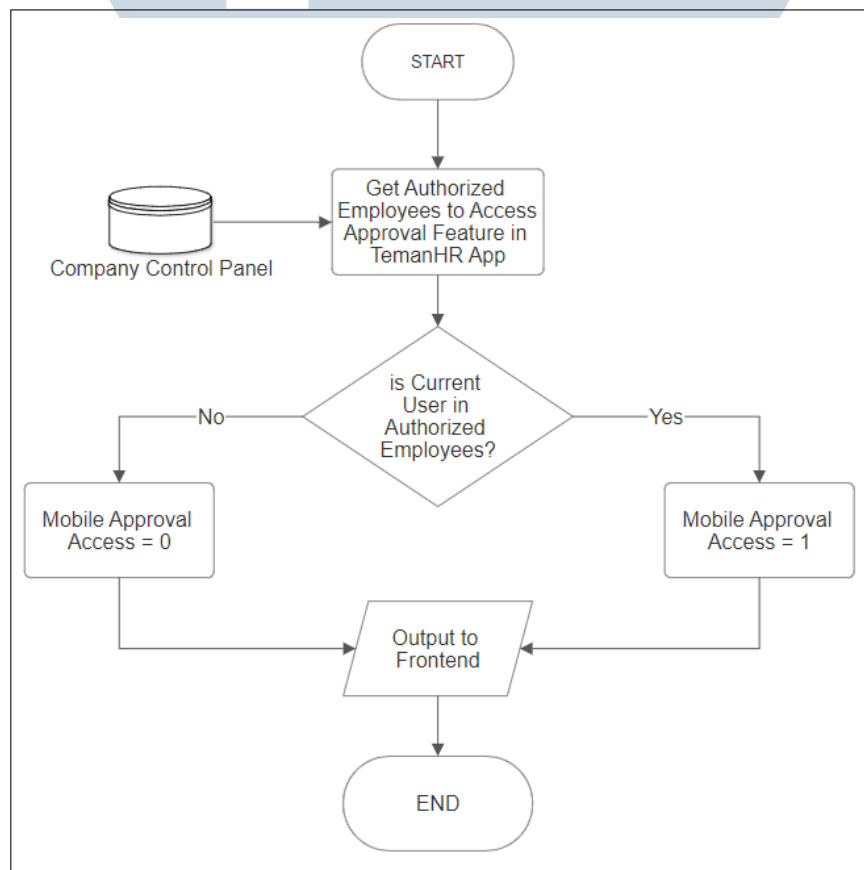
1. User Requirement

Adapun *user requirement* untuk fitur akses pengajuan *mobile* yaitu sebagai berikut.

- (a) *User* yang telah ditentukan harus langsung memiliki akses untuk menggunakan fitur pengajuan *mobile* ketika membuka aplikasi *TemanHR*.
- (b) *User* yang berwenang harus bisa menentukan siapa karyawan *HR* yang bisa menggunakan fitur ini.

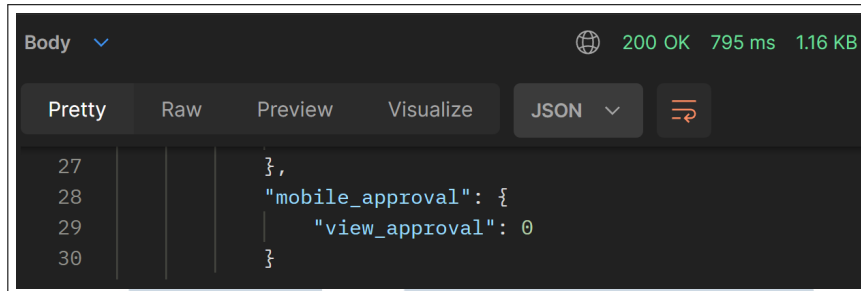
2. Flowchart

Gambar 3.14 merupakan *flowchart* yang mencakup beberapa proses. Dimulai dengan pengambilan karyawan yang terdaftar untuk memiliki akses pengajuan *mobile*. Selanjutnya dilakukan pengecekan apakah pengguna aplikasi ada di dalam daftar tersebut. Jika iya, maka akan memberikan informasi ke *frontend software developer* bahwa pengguna memiliki akses pengajuan *mobile*.



Gambar 3.14. Flowchart Akses Pengajuan Mobile

3. Implementasi



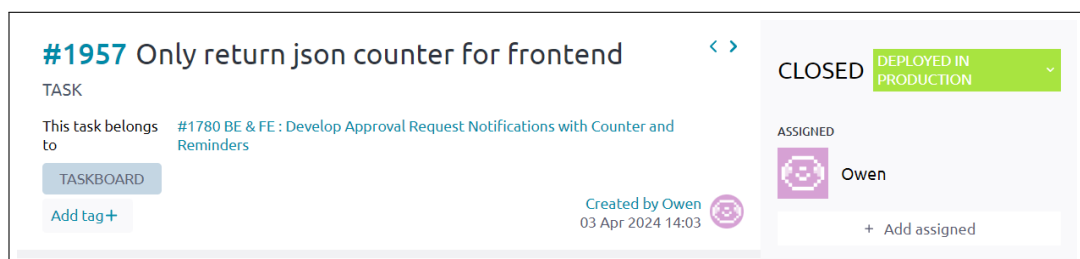
```
Body 200 OK 795 ms 1.16 KB
Pretty Raw Preview Visualize JSON
27 },
28   "mobile_approval": {
29     "view_approval": 0
30   }
```

Gambar 3.15. JSON Akses Pengajuan Mobile

Gambar 3.15 merupakan potongan *JSON* yang mengirimkan informasi pengguna kepada *frontend software developer* pada saat pengguna masuk kedalam aplikasi *TemanHR*. Pesan pada *JSON* tersebut memberikan informasi kepada frontend apabila pengguna memiliki akses pengajuan mobile atau tidak. Jika pesan "view_approval" adalah 0 maka tidak ada akses pengajuan mobile.

D. Fitur Notifikasi Penghitung Pengajuan Terbuka

Fitur notifikasi penghitung pengajuan terbuka adalah fitur yang menghitung total pengajuan yang belum disetujui atau ditolak oleh karyawan *HR*. Tujuan dari fitur ini adalah untuk memberikan informasi yang simpel terkait pengajuan yang belum disetujui atau ditolak. Informasi yang diberikan adalah total dari semua dokumen yang perlu dikonfirmasi oleh karyawan *HR*. Gambar 3.16 adalah bukti bahwa fitur ini telah mendapatkan persetujuan dari *product owner* dan sudah sesuai dengan *user requirement* yang sudah ditentukan sebelumnya.



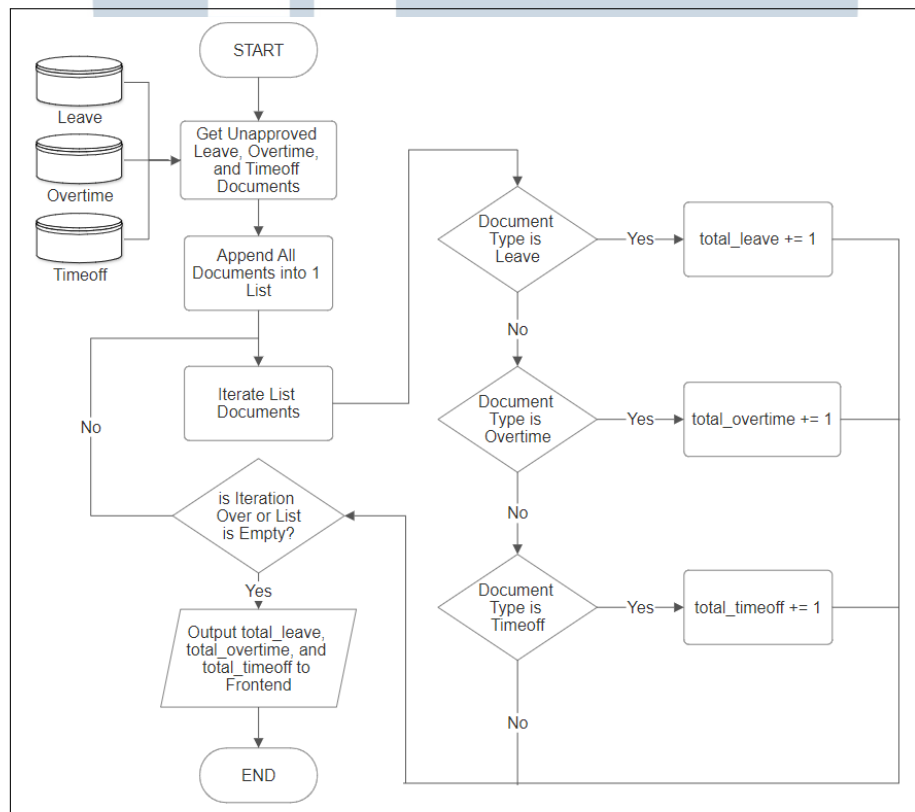
Gambar 3.16. Persetujuan Fitur Notifikasi Penghitung Pengajuan Terbuka

1. User Requirement

Adapun *user requirement* untuk fitur notifikasi penghitung pengajuan terbuka yaitu sebagai berikut.

- (a) Sistem harus mengirim notifikasi otomatis setiap kali ada pengajuan dokumen yang baru.
- (b) Sistem harus memberikan total pengajuan dokumen yang belum di konfirmasi.

2. Flowchart

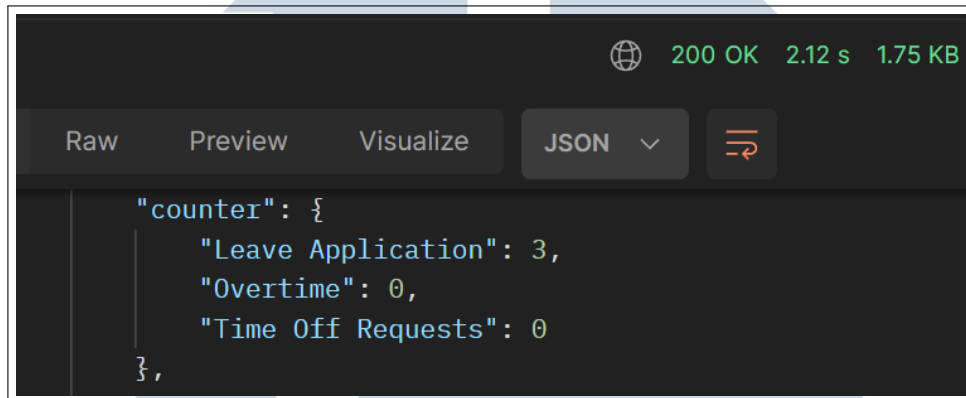


Gambar 3.17. Flowchart Notifikasi Penghitung Pengajuan

Gambar 3.17 merupakan *flowchart* yang mencakup beberapa proses. Dimulai dari pengambilan semua dokumen cuti, lembur, dan *timeoff* yang terbuka dan butuh konfirmasi dari karyawan *HR*. Selanjutnya, semua dokumen tersebut dijadikan satu daftar. Selanjutnya, diberikan nilai variabel total dokumen awal yaitu nol. Selanjutnya, dilakukan pengulangan penghitungan total dokumen dari daftar yang sudah dibuat. Jika daftar

tersebut kosong, maka pengulangan akan berhenti dan memberikan informasi kepada *frontend software developer* total notifikasi yang terhitung.

3. Implementasi



Gambar 3.18. JSON Notifikasi Penghitung Pengajuan

Gambar 3.18 merupakan potongan *JSON* yang memberikan informasi total dari pengajuan yang perlu dikonfirmasi. Tujuan dari informasi ini adalah untuk memberikan total pengajuan yang perlu dikonfirmasi kepada *frontend software developer* untuk dijadikan fitur notifikasi penghitungan pengajuan pada aplikasi *TemanHR*.

E. Fitur Notifikasi Peningat Absen

Fitur notifikasi peningat absen adalah fitur yang mengirimkan notifikasi kepada perangkat *mobile* setiap karyawan untuk melakukan absensi pada notifikasi berjalan. Fitur ini berjalan selama 15 menit setiap hari untuk menyesuaikan perbedaan jam kerja perusahaan. Tujuan dari fitur ini adalah untuk mengingatkan para karyawan supaya tidak ada kekosongan absensi. Notifikasi akan dikirim secara otomatis oleh sistem pada saat jam perusahaan buka dan tutup setiap hari. Gambar 3.19 adalah bukti bahwa fitur ini telah mendapatkan persetujuan dari *product owner* dan sudah sesuai dengan *user requirement* yang sudah ditentukan sebelumnya.



Gambar 3.19. Persetujuan Fitur Notifikasi Pengingat Absen

1. User Requirement

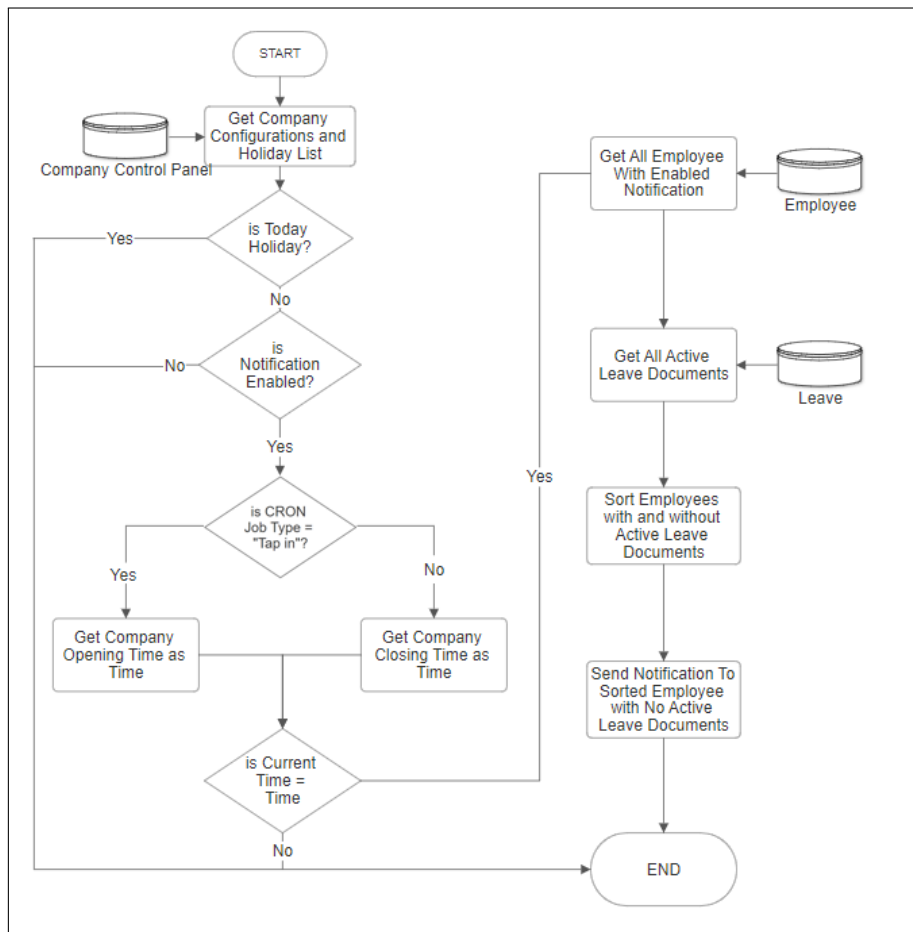
Adapun *user requirement* untuk fitur validasi minimal jam kerja sebelum absen pulang yaitu sebagai berikut.

- (a) Sistem harus mengirim notifikasi otomatis setiap hari pada saat jam buka dan jam tutup perusahaan.
- (b) Sistem harus mengirim notifikasi kepada karyawan yang tidak memiliki dokumen cuti aktif saja.
- (c) Notifikasi harus memiliki format yang jelas dan informatif.

2. Flowchart

Gambar 3.20 merupakan *flowchart* yang mencakup beberapa proses. Dimulai dengan pengambilan data pengaturan perusahaan dan daftar hari libur. Selanjutnya dilakukan pengecekan apakah hari adalah hari libur. Jika hari ini libur maka notifikasi tidak akan berjalan. Selanjutnya dilakukan pengecekan apakah fitur notifikasi dinyalakan pada pengaturan perusahaan. Jika iya, maka notifikasi tidak akan berjalan. Selanjutnya dilakukan pengecekan tipe *CRON Job*. Jika tipe *CRON Job* adalah "Tap in", maka waktu yang dibandingkan adalah waktu perusahaan buka, namun jika "Tap out", maka waktu yang dibandingkan adalah waktu perusahaan tutup.

Setelah itu, dilakukan pengambilan semua karyawan dan dokumen cuti yang aktif pada hari notifikasi berjalan. Selanjutnya dilakukan sortir untuk karyawan yang memiliki dan tidak memiliki dokumen cuti yang aktif pada hari itu. Selanjutnya dilakukan pengiriman notifikasi untuk karyawan yang tidak memiliki dokumen cuti yang aktif pada hari itu.



Gambar 3.20. Flowchart Notifikasi Peningkat Absen

3. Implementasi

Gambar 3.21 merupakan notifikasi dari fitur notifikasi peningkat absen. Notifikasi tersebut memiliki judul dan badan yang bergantung pada tipe notifikasi yang dikirim.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.21. Notifikasi Peningat Absen

3.4.2 Pengembangan Sistem ERPNext Klien

Sistem ERPNext adalah sistem Enterprise Resource Planning (ERP) yang berbasis web dan open-source. Sistem ini, yang dikembangkan oleh Frappe Technologies, memberikan dukungan untuk berbagai fungsi bisnis termasuk akuntansi, penjualan, dan pembelian [6]. Salah satu klien PT Sumber Inovasi Informatika telah memilih layanan perancangan dan perkembangan sistem ERPNext yang disesuaikan dengan alur bisnis klien. Saat bergabung dengan proyek ERPNext klien, tahap yang sedang ditempuh adalah tahap development. Kontribusi yang sudah diberikan kepada proyek ini adalah sebagai berikut.

A. Fitur Validasi Perbandingan Nilai Customer's Purchase Order dan Nilai Total Barang pada Sales Order

Fitur validasi perbandingan nilai *customer's purchase order* dan nilai total barang pada *sales order* adalah fitur yang melakukan pengecekan kedua nilai tersebut. Tujuan dari fitur tersebut adalah untuk mencegah adanya perbedaan antara nilai *customer's purchase order* dan nilai total barang pada *sales order* pada saat pembuatan dokumen *invoice proposal*. Gambar 3.22 adalah bukti bahwa fitur ini telah mendapatkan persetujuan dari *product owner* dan sudah sesuai dengan *user requirement* yang sudah ditentukan sebelumnya.



Gambar 3.22. Persetujuan Fitur Validasi Perbandingan Nilai Customer's Purchase Order dan Nilai Total Barang pada Sales Order

1. User Requirement

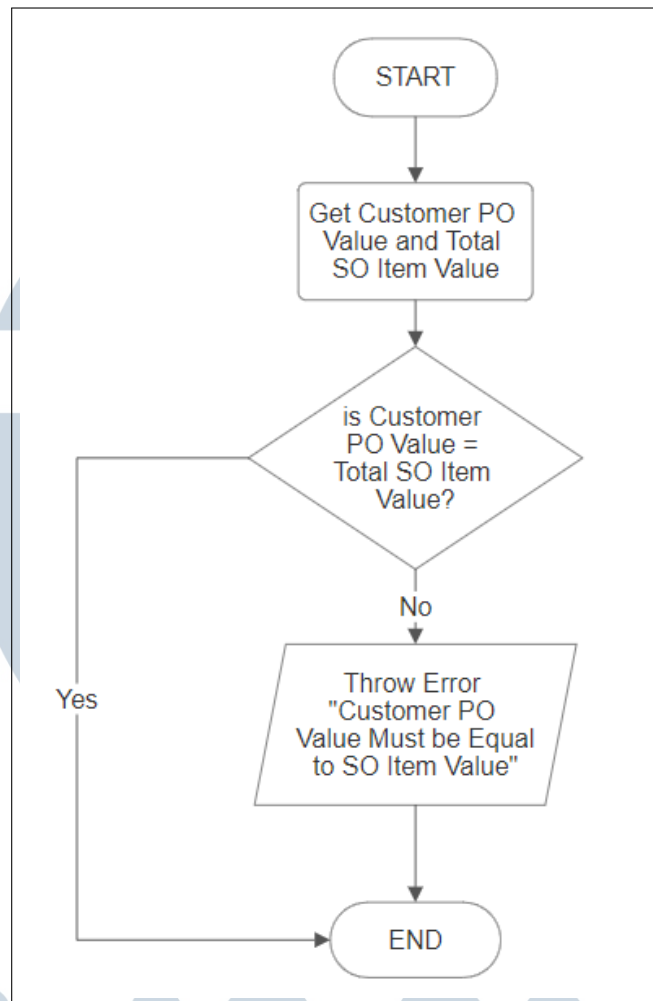
Adapun *user requirement* untuk fitur validasi minimal jam kerja sebelum absen pulang yaitu sebagai berikut.

- (a) User harus tidak diperbolehkan untuk melakukan submisi jika nilai *customer's purchase order* dan nilai total barang pada *sales order* berbeda.
- (b) User harus diberikan pesan yang informatif ketika tidak diperbolehkan untuk melakukan submisi.

2. Flowchart

Gambar 3.23 merupakan *flowchart* yang mencakup beberapa proses. Dimulai dengan pengambilan nilai *customer purchase order* dan nilai total barang *sales order*. Jika kedua nilai tersebut berbeda maka sistem akan memberikan *output* eror dan membatalkan pembuatan dokumen *invoice proposal*.

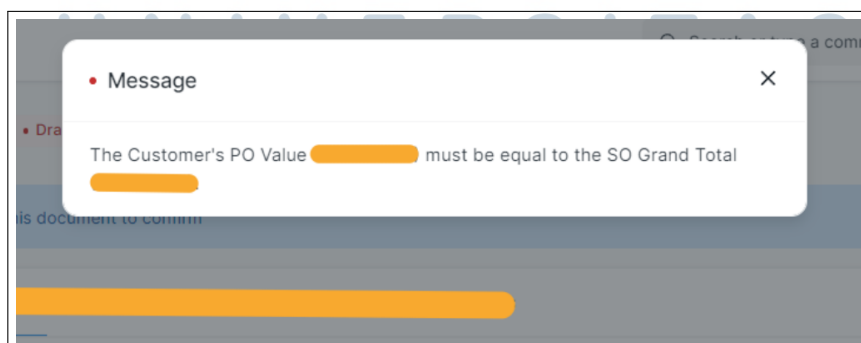
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.23. Flowchart Validasi Perbandingan

3. Implementasi

Gambar 3.24 merupakan potongan *output* eror yang muncul ketika nilai *customer purchase order* dan nilai total barang *sales order* berbeda.



Gambar 3.24. Validasi Perbandingan

B. Fitur Deteksi Status Proyek dari Persentase Penyelesaian

Fitur deteksi status proyek dari penyelesaian proyek merupakan fitur yang melakukan pendeteksian status dari persentase penyelesaian pada suatu proyek. Tujuan dari fitur ini adalah untuk mengubah proyek yang sudah seratus persen selesai. Gambar 3.25 adalah bukti bahwa fitur ini telah mendapatkan persetujuan dari *product owner* dan sudah sesuai dengan *user requirement* yang sudah ditentukan sebelumnya.



Gambar 3.25. Persetujuan Fitur Deteksi Status Proyek dari Persentase Penyelesaian

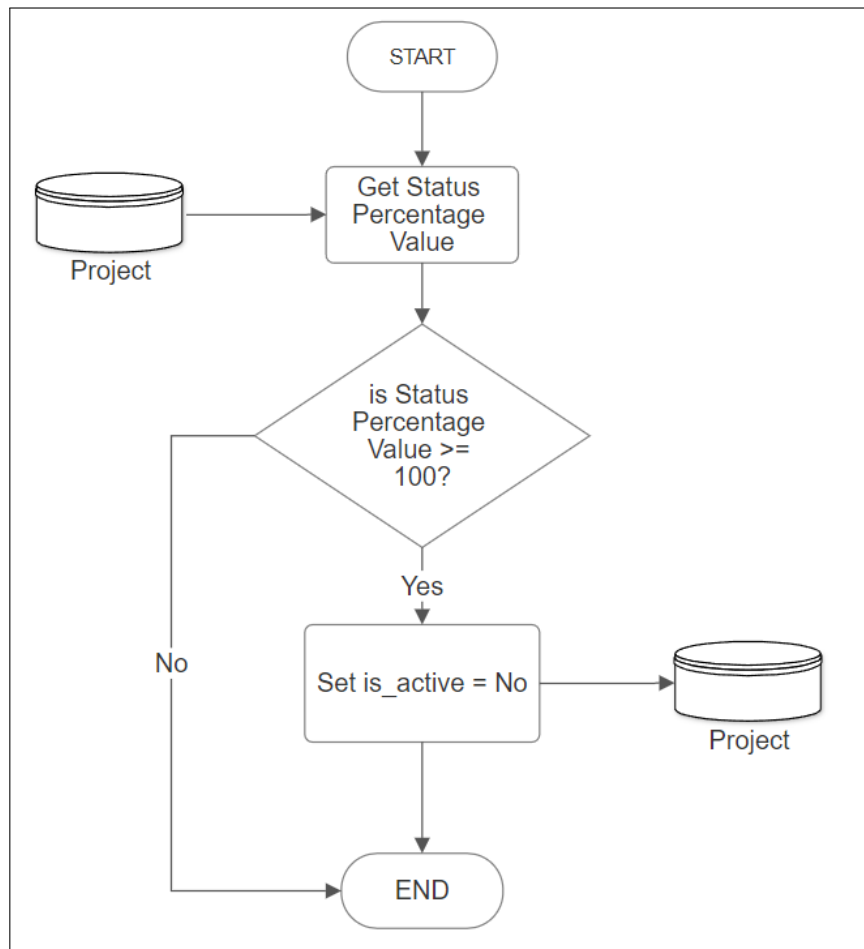
1. User Requirement

Adapun *user requirement* untuk fitur validasi minimal jam kerja sebelum absen pulang yaitu sebagai berikut.

- (a) Sistem harus dapat mendeteksi secara otomatis persentase penyelesaian proyek.
- (b) Sistem harus mengubah status proyek secara otomatis ketika proyek sudah seratus persen selesai.

2. Flowchart

Gambar 3.26 merupakan *flowchart* yang mencakup beberapa proses. Dimulai dengan pengambilan nilai persentase proyek. Selanjutnya, dilakukan pengecekan apakah persentase tersebut adalah seratus persen. Jika iya, maka status proyek akan tidak aktif.



Gambar 3.26. Flowchart Deteksi Status Proyek dari Persentase Penyelesaian

3. Implementasi

Gambar 3.27 merupakan potongan daftar proyek yang memiliki persentase penyelesaian yang seratus dan nol. Jika seratus maka proyek akan tidak aktif, jika nol maka proyek masih aktif.

<input type="checkbox"/>	♥ Project Name	Status	% Completed	Parent Project	Is Active
<input type="checkbox"/>	♥ [Orange Bar]	• On Progress	[Grey Bar]	[Orange Bar]	• Yes
<input type="checkbox"/>	♥ Project Name	Status	% Completed	Parent Project	Is Active
<input type="checkbox"/>	♥ [Orange Bar]	• Approved By	[Green Bar]	[Orange Bar]	• No

Gambar 3.27. Validasi Perbandingan

C. Fitur Penampilan Tipe Mata Uang Payment Entry pada Laporan Proyek

Fitur penampilan tipe mata uang *payment entry* pada laporan proyek merupakan fitur yang memberikan informasi tipe mata uang dokumen *payment entry* pada laporan proyek. Tujuan dari fitur ini adalah untuk memberikan informasi yang lebih detail terkait mata uang yang digunakan pada dokumen *payment entry*. Gambar 3.28 adalah bukti bahwa fitur ini telah mendapatkan persetujuan dari *product owner* dan sudah sesuai dengan *user requirement* yang sudah ditentukan sebelumnya.



Gambar 3.28. Persetujuan Fitur Penampilan Tipe Mata Uang Payment Entry pada Laporan Proyek

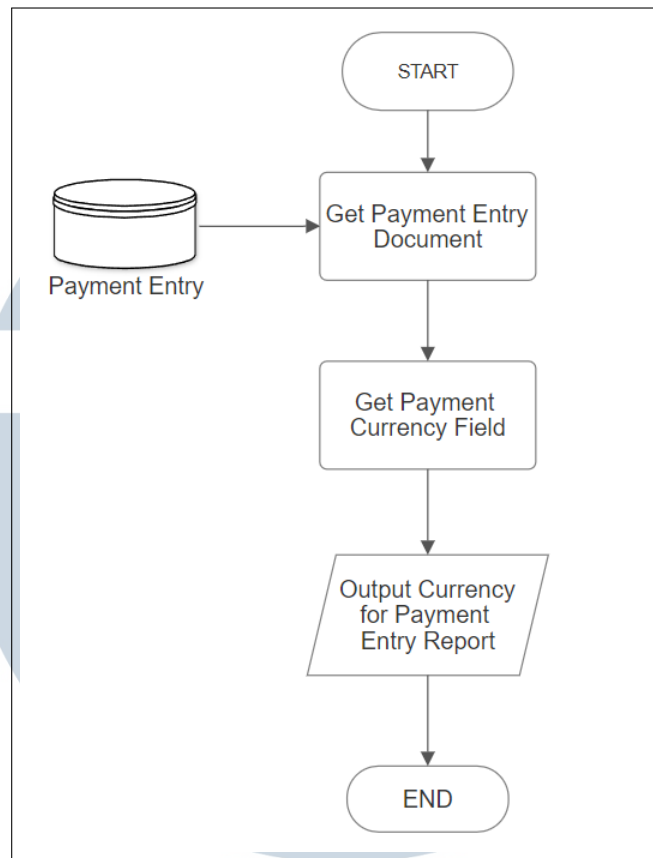
1. User Requirement

Adapun *user requirement* untuk fitur validasi minimal jam kerja sebelum absen pulang yaitu sebagai berikut.

- (a) Sistem harus mendeteksi mata uang yang dipakai untuk pembayaran.
- (b) Sistem harus menampilkan mata uang yang dipakai untuk pembayaran pada laporan proyek.

2. Flowchart

Gambar 3.29 merupakan *flowchart* yang mencakup beberapa proses. Dimulai dengan pengambilan nilai persentase proyek. Selanjutnya, dilakukan pengecekan apakah persentase tersebut adalah seratus persen. Jika iya, maka status proyek akan tidak aktif.



Gambar 3.29. Flowchart Penampilan Tipe Mata Uang *Payment Entry* pada Laporan Proyek

3. Implementasi

Gambar 3.27 merupakan potongan laporan proyek yang menampilkan tipe mata uang dan total uang pada *payment entry*.

Settlement Currency	Settlement Amount
IDR	

Gambar 3.30. Penampilan Tipe Mata Uang *Payment Entry* pada Laporan Proyek

3.5 Kendala dan Solusi yang Ditemukan

Kendala yang ditemukan selama kerja magang di PT Sumber Inovasi Informatika yaitu sebagai berikut.

1. Tantangan dalam proses pengembangan dan pembuatan kode yang berhubungan dengan sistem *ERPNext* karena sistem *ERPNext* membutuhkan pemahaman mendalam tentang akuntansi seperti debit/kredit, penjualan, dan pemesanan.
2. Tantangan dalam menyelesaikan tugas karena kesalahpahaman antara *backend software developer* dan *product owner* dari tiket yang diberikan oleh *product owner*.

Solusi yang ditemukan untuk kendala-kendala tersebut selama magang di PT Sumber Inovasi Informatika yaitu sebagai berikut.

1. Mengikuti kelas akuntansi internal yang disediakan oleh PT Sumber Inovasi Informatika setiap minggu pada hari Kamis jam tiga selama enam minggu.
2. Meminta penjelasan yang lebih detail kepada *product owner* untuk memastikan kedua pihak melihat visi yang sama pada suatu tiket.

